

Amendments to the Specification:

Please replace the paragraph on page 8, line 30, through page 9, line 2, with the following amended paragraph.

The node then determines if it is still in poll-request mode[.]. ~~if~~ **If** not, it proceeds to transition Step D, if it is still under contention[.]. ~~for~~ **For** instance, multiple clients have just requested to transmit data to it in this PRP cycle, then the node stays in PRP mode. Before starting a new PRP cycle, the node can attempt to send data that it has that was not destined to any of the requesting clients. (This data would have been sent in Step C4). For instance, some of the data that a requesting client has just sent may be forwarded further along in the network.

Please delete line 4 on page 9.

DESCRIPTION OF FIELDS IN PACKETS

Please replace the paragraph on page 10, lines 1-8, with the following amended paragraph.

Referring to Fig. 5C, it then continues by listening during one or more reserved minislots for a Poll Packet (Fig. 3B) (Step 5C-1) and if it hears a Poll Packet (5C-2) responds as before by recording the Poll Packet (5C-3), incrementing a second counter (5C-4) and tests to see if the second counter value equals the number of reserved minislots (5C-5), repeating the process until it does equal. When it ~~continues~~ **does equal**, it constructs a Contention Resolution Packet (Fig. 3C) (Step 5C-6), which is used to carry the sending transmit start time to clients and pollers. It then broadcasts this Contention Resolution Packet (5C-7).

Please replace the paragraph on page 10, lines 9-17, with the following amended paragraph.

Referring to Fig. 5D, the node next sets up and triggers a timer to set the maximum listening time for the end of an expected variable length data packet (Fig. 2 at C4) (Step 5D-1). This could be a timer for each data packet or it could be a timer for all expected packets in a sequence. The node then listens for the Data Packets (5D-2) and if it doesn't hear one (5D-3) it checks for time expiration (5D-6) and either repeats or times out. If it hears a Data Packet, it records the Data Packet (5D-4) and checks to see if this is the last Data Packet to listen for (5D-5). Once it has completed listening, it sends the Designated Data Packets directed to the clients (5D-7), then checks all received packets for correctness, deleting those that are incorrect (5D-8) before continuing.

Please replace the paragraph on page 11, lines 1-5, with the following amended paragraph.

Until the timer expires, the client listens for the **Broadcast Poll Request Packet** (6A-5) and if it doesn't hear it, continues with other work (6A-10). When it hears the Broadcast Poll Request Packet, it checks for a reserved contention minislot for itself (6A-7). Upon finding none it sends a Poll Packet in any random contention minislot (6A-8) and continues. Otherwise it sends a Poll Packet in the specified reserved minislot (6A-9) and continues.

Please replace the paragraph on page 11, lines 6-11, with the following amended paragraph.

Referring to Fig. 6B, the client then listens for the Broadcast ~~response~~ **Resolution** Packet (6B-1) and finding none (6B-2) continues; otherwise it checks to see if its own data was requested (6B-3) and if so, sends packets at the specified time (6B-4). It then determines if the

data packets are to be sent to it (6B-5); if not it continues; otherwise it listens for the data packets at the specified time (6B-6). It also tests for the reception of data packets designated for it (Step 6B-7) and proceeds with the processes.

Please replace the paragraph on page 11, lines 12-20, with the following amended paragraph.

Referring to 6C, if Data Packets are heard, it prepares a Receive Acknowledgment Packet(6C-1) and listens for a broadcast acknowledgment from the server (6C-2). Continuing it checks for whether it heard the Broadcast Acknowledgment of the sent packets (6C-3) and if heard, deletes the acknowledged packets from its own send queue (6C-4) and sends the receive acknowledgment packet (6C-5). Checking to see if any packets are left to be sent ~~by~~ to the server (6C-6), if yes it reverts to the timer setup (6A-2) to repeat the process. If not, it checks for any further expected data to transfer (6C-7) and either reverts as above or if nothing further is expected, leaves the PRPstate with the server (6C-8).